

**AMENDMENTS TO THE CLAIMS**

1 1.-15. (Canceled)

1 16. (Currently Amended) A method for routing or switching data packets, comprising the

2 computer-implemented steps of:

3 receiving a data packet at an input interface on a router or switch;

4 looking up information in the header of said data packet in an expanded M-trie data

5 structure, wherein said expanded M-trie data structure is organized as a

6 multi-level tree including a root node, inferior nodes, and terminal nodes,

7 wherein each node stores values for an address and an opcode, wherein said

8 opcode specifies:

9 a particular field of a plurality of fields in the header of said data packet; and

10 an operation that is to be performed on the data stored in said particular field,

11 wherein said operation is one of a plurality of operations that said

12 opcode can specify; and

13 terminating said step of looking up information.

1 17. (Canceled)

1 18. (Canceled)

1 19. (Previously Presented) A method as in claim 16, wherein said address includes the

2 address of a node in said expanded M-trie data structure that is to be traversed.

1 20. (Original) A method as in claim 16, wherein said expanded M-trie data structure

2 includes a set of access control parameters.

- 1 21. (Previously Presented) A method as in claim 16, wherein said expanded M-trie data  
2 structure includes a set of Quality of Service (QoS) parameters.
- 1 22. (Previously Presented) A method as in claim 16, wherein said expanded M-trie data  
2 structure includes a set of Class of Service (CoS) parameters.
- 1 23. (Previously Presented) A method as in claim 16, wherein said nodes include opcodes for  
2 demultiplexing, opcodes for matching, and opcodes for hashing.
- 1 24. (Previously Presented) A method as in claim 23, wherein said opcodes for  
2 demultiplexing include instructions to demultiplex into branches of said expanded  
3 M-trie data structure based on contents of a byte of said packet header that is being  
4 read.
- 1 25. (Previously Presented) A method as in claim 23, wherein said opcodes for matching  
2 include instructions to compare the contents of a given byte of the flow label to given  
3 node data.
- 1 26. (Previously Presented) A method as in claim 23, wherein said opcodes for hashing  
2 include instructions to hash into different M-trie plus branches based on the contents  
3 of a given byte in said packet header.
- 1 27. (Canceled)
- 1 28. (Currently Amended) An apparatus for routing or switching data packets, comprising a  
2 device that performs a method comprising:  
3 storing in memory an M-trie data structure, said data structure organized as a multi-  
4 level tree having a set of nodes, including a root node, inferior nodes and  
5 terminal nodes, wherein each node stores values for an address and an  
6 opcode, wherein said opcode specifies:

7 a particular field of a plurality of fields of data packet headers; and  
8 an operation that is to be performed on the data stored in said particular field,  
9 wherein said operation is one of a plurality of operations that said  
10 opcode can specify;  
11 receiving a data packet at an input interface on a router or switch, wherein the data  
12 packet includes information in at least a header with at least a field that is  
13 used by said M-trie data structure to indicate an action for said device to  
14 perform in order to select a leaf associated with said M-trie data structure;  
15 looking up the information, wherein the looking up includes performing the action;  
16 and  
17 routing said data packet at one or more output interfaces on said router or said  
18 switch.

1 29. (Currently Amended) A method for routing or switching data packets, comprising the  
2 computer-implemented steps of:  
3 storing in memory an M-trie data structure, said data structure organized as a multi-  
4 level tree having a set of nodes, including a root node, inferior nodes and  
5 terminal nodes, wherein each node stores values for an address and an  
6 opcode, wherein said opcode specifies:  
7 a particular field of a plurality of fields of data packet headers; and  
8 an operation that is to be performed on the data stored in said particular field,  
9 wherein said operation is one of a plurality of operations that said  
10 opcode can specify;  
11 receiving a data packet at an input interface on a router or switch, wherein the data  
12 packet includes information in at least a header with at least a field that is  
13 used by said M-trie data structure to indicate an action for a router to perform  
14 in order to select a leaf associated with said M-trie data structure; and  
15 looking up the information, wherein the looking up includes performing the action.

1 30. (Currently Amended) A memory storing a program for performing a method for routing  
2 or switching data packets, comprising:

3 storing in memory an M-trie data structure, said data structure organized as a multi-  
4 level tree having a set of nodes, including a root node, inferior nodes and  
5 terminal nodes, wherein each node stores values for an address and an  
6 opcode, wherein said opcode specifies:  
7 a particular field of a plurality of fields of data packet headers; and  
8 an operation that is to be performed on the data stored in said particular field,  
9 wherein said operation is one of a plurality of operations that said  
10 opcode can specify;  
11 receiving a data packet at an input interface on a router or switch, wherein the data  
12 packet includes information in at least a header with at least a field that is  
13 used by said M-trie data structure to indicate an action for a router to perform  
14 in order to select a leaf associated with said M-trie data structure;  
15 looking up the information, wherein the looking up includes performing the action;  
16 and  
17 routing said data packet at one or more output interfaces on said router or said  
18 switch.

1 31. (Canceled)

1 32. (Previously Presented) A memory as in claim 30, wherein said address includes an  
2 address of a node in said M-trie data structure that is to be traversed.

1 33. (Previously Presented) A memory as in claim 30, wherein said M-trie data structure  
2 includes a set of access control parameters.

1 34. (Previously Presented) A memory as in claim 30, wherein said M-trie data structure  
2 includes a set of Quality of Service (QoS) parameters.

1 35. (Previously Presented) A memory as in claim 30, wherein said expanded M-trie data  
2 structure includes a set of Class of Service (CoS) parameters.

- 1 36. (Previously Presented) A memory as in claim 30 wherein at least one of the root node,  
2 inferior nodes, or the terminal node includes an opcode for demultiplexing, an  
3 opcode for matching, and an opcode for hashing.
- 1 37. (Previously Presented) A memory as in claim 36 wherein said opcode for  
2 demultiplexing includes instructions to demultiplex into branches of the M-trie data  
3 structure based on contents of a byte of said packet header.
- 1 38. (Previously Presented) A memory as in claim 36, wherein said opcode for matching  
2 includes instructions to compare the contents of a given byte of a flow label to given  
3 node data.
- 1 39. (Previously Presented) A memory as in claim 36, wherein said opcode for hashing  
2 includes instructions to hash into different branches the M-trie data structure based  
3 on the contents of a given set of bytes in said packet header.
- 1 40. (Previously Presented) A method as recited in Claim 16, further comprising routing  
2 said data packet at one or more output interfaces on said router or said switch.
- 1 41. (Previously Presented) A method as recited in Claim 16, further comprising  
2 determining, based on one or more Access Control List (ACL) criteria stored in said  
3 expanded M-trie data structure, whether to drop or forward said data packet.
- 1 42. (Previously Presented) A method as recited in Claim 41, wherein determining  
2 whether to drop or forward said data packet comprises matching said information in  
3 the header of said data packet to the one or more ACL criteria stored in said  
4 expanded M-trie data structure.
- 1 43. (Previously Presented) A method as recited in Claim 41, wherein said one or more  
2 ACL criteria include at least one of a source address, destination address, and upper-  
3 layer protocol information.

1 44. (Previously Presented) A method as recited in Claim 41, wherein said one or more  
2 ACL criteria are stored in a sub-tree of said expanded M-trie data structure.

1 45. (Previously Presented) A method as recited in Claim 29, further comprising routing  
2 said data packet at one or more output interfaces on said router or said switch.

1 46. (Previously Presented) A method as recited in Claim 29, further comprising  
2 determining, based on one or more Access Control List (ACL) criteria stored in said  
3 M-trie data structure, whether to drop or forward said data packet.

1 47. (Previously Presented) A method as recited in Claim 46, wherein determining  
2 whether to drop or forward said data packet comprises matching said information to  
3 the one or more ACL criteria stored in said M-trie data structure.

1 48. (Previously Presented) A method as recited in Claim 46, wherein said one or more  
2 ACL criteria include at least one of a source address, a destination address, and  
3 upper-layer protocol information.

1 49. (Previously Presented) A method as recited in Claim 46, wherein said one or more  
2 ACL criteria are stored in a sub-tree of said M-trie data structure.